

```
BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
BBBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBB      BBB      AAA      AAA      SSS      SSS      RRR      RRR      TTT      TTT      LLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLLLLLLLLLLLLLLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLLLLLLLLLLLLLLL
BBBBBBBBBBBBBB      AAA      AAA      SSSSSSSSSSSS      RRR      RRR      TTT      TTT      LLLLLLLLLLLLLLLL
```

```
BBBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      YY      YY      SSSSSSSS
BBBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      YY      YY      SSSSSSSS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BBBBBBBBB      AA      AA      SSSSSS      SSSSSS      YY      YY      SSSSSS
BBBBBBBBB      AA      AA      SSSSSS      SSSSSS      YY      YY      SSSSSS
BB      BB      AAAAAAAAAA      SS      SS      YY      YY      SS      SS
BB      BB      AAAAAAAAAA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SS      SS      YY      YY      SS      SS
BB      BB      AA      AA      SSSSSSSS      SSSSSSSS      YY      YY      SSSSSSSS
BBBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      YY      YY      SSSSSSSS
BBBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      YY      YY      SSSSSSSS
```

....  
....  
....  
....

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

```
1 0001 0 MODULE BAS$SYS ( ! RSTS SYS Function
2 0002 0 IDENT = '1-014' ! File: BAS$SYS.B32 Edit: PL1014
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: VAX-11 BASIC RSTS COMPATABILITY
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains the RSTS-compatible SYS function.
36 0036 1 Some of the more complex functions do calls to BPA routines.
37 0037 1
38 0038 1 ENVIRONMENT: VAX-11 User Mode
39 0039 1
40 0040 1 AUTHOR: John Sauter, CREATION DATE: 01-OCT-1979
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original.
45 0045 1 1-002 - Define BPAS_HISEG as 0, to disable the checking for the end
46 0046 1 of the compatibility-mode high segment, and allocate
47 0047 1 some global cells for ASSIGN/DEASSIGN. JBS 02-OCT-1979
48 0048 1 1-003 - Instead of passing the XRB to the message send/receive code,
49 0049 1 pass the relevant fields of the XRB. Also, BPAS_HISEG is
50 0050 1 no longer needed. JBS 04-OCT-1979
51 0051 1 1-004 - Debug message send/receive. JBS 05-OCT-1979
52 0052 1 1-005 - Continue debugging message send/receive. JBS 07-OCT-1979
53 0053 1 1-006 - Fix some errors in calling terminal functions. JBS 12-OCT-1979
54 0054 1 1-007 - Handle short parameter strings correctly. JBS 17-OCT-1979
55 0055 1 1-008 - Add core common. JBS 03-DEC-1979
56 0056 1 1-009 - Add FSS. JBS 04-DEC-1979
57 0057 1 1-010 - Add setting priority. JBS 04-DEC-1979
```



BASSSYS  
1-014

K 3  
16-Sep-1984 01:16:51 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 11:56:41 [BASRTL.SRC]BASSYS.B32;1

Page 2  
(1)

```
: 58      0058 1 : 1-011 - Make "exit with no prompt" exit more quietly, fix a typo in putting
: 59      0059 1 :      into core common, and make ODI submode give an error message, since
: 60      0060 1 :      there isn't time to make it work right for this release. JBS 26-FEB-1980
: 61      0061 1 : 1-012 - Put in small send and receive. All four functions. FM 24-FEB-81.
: 62      0062 1 : 1-013 - LIB$STOP should be declared EXTERNAL. PL 20-Nov-81
: 63      0063 1 : 1-014 - Call BASS$STOP to signal errors instead of BASS$STOP_10. PLL 16-Jun-1982
: 64      0064 1 : --
: 65      0065 1 :
: 66      0066 1 : <BLF/PAGE>
```

```
68 0067 1 |
69 0068 1 | SWITCHES:
70 0069 1 |
71 0070 1 |
72 0071 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
73 0072 1 |
74 0073 1 |
75 0074 1 | LINKAGES:
76 0075 1 |
77 0076 1 |
78 0077 1 | REQUIRE 'RTLIN:OTSLNK'; ! I/O Linkages
79 0506 1 |
80 0507 1 |
81 0508 1 | TABLE OF CONTENTS:
82 0509 1 |
83 0510 1 |
84 0511 1 | FORWARD ROUTINE
85 0512 1 |     BASSSYS : NOVALUE, ! Do a SYS() function
86 0513 1 |     BASS$UUD : NOVALUE; ! Do a UUD sub-function
87 0514 1 |
88 0515 1 |
89 0516 1 | INCLUDE FILES:
90 0517 1 |
91 0518 1 |
92 0519 1 | REQUIRE 'RTLML:OTSLUB'; ! LUB definitions
93 0659 1 |
94 0660 1 | REQUIRE 'RTLIN:BPASTRUCT'; ! Structuring definitions
95 0751 1 |
96 0752 1 | REQUIRE 'RTLIN:BPAFQBDEF'; ! FIRQB definitions
97 0876 1 |
98 0877 1 | REQUIRE 'RTLIN:BPAFSBDEF'; ! FSB definitions
99 1008 1 |
100 1009 1 | REQUIRE 'RTLIN:BPAFUNDEF'; ! Function definitions
101 1259 1 |
102 1260 1 | REQUIRE 'RTLIN:RTLPSECT'; ! Macros for defining psects
103 1355 1 |
104 1356 1 | LIBRARY 'RTLSTARLE'; ! System symbols
105 1357 1 |
106 1358 1 |
107 1359 1 | MACROS:
108 1360 1 |
109 1361 1 |     NONE
110 1362 1 |
111 1363 1 | EQUATED SYMBOLS:
112 1364 1 |
113 1365 1 |     NONE
114 1366 1 |
115 1367 1 | PSECTS:
116 1368 1 |
117 1369 1 | DECLARE_PSECTS (BAS); ! Declare psects for BASS$ facility
118 1370 1 |
119 1371 1 | OWN STORAGE:
120 1372 1 |
121 1373 1 |     NONE
122 1374 1 |
123 1375 1 | EXTERNAL REFERENCES:
124 1376 1 |
```

```
125 1377 1
126 1378 1 EXTERNAL ROUTINE
127 1379 1   LIB$STOP : NOVALUE,      ! Signal a fatal error
128 1380 1   STR$COPY_DX,        ! Copy a string by descriptor
129 1381 1   STR$COPY_R,        ! Copy a string by reference
130 1382 1   STR$FREE_DX,       ! Free a string
131 1383 1   STR$CONCAT,        ! Concatenate strings
132 1384 1   LIB$GET_COMMON,    ! Fetch from process common
133 1385 1   LIB$PUT_COMMON,    ! Store in process common
134 1386 1   BASS$STOP : NOVALUE, ! Signal fatal error
135 1387 1   BASS$CTRL0 : NOVALUE, ! Clear control 0
136 1388 1   BASS$NOECHO : NOVALUE, ! Turn off echoing
137 1389 1   BASS$ONECHR : NOVALUE, ! Enter ODT submode
138 1390 1   BASS$CANTYPAHEAD : NOVALUE, ! Cancel type-ahead
139 1391 1   BASS$CTRLC : NOVALUE, ! Enable control-C trapping
140 1392 1   BASS$ECHO : NOVALUE, ! Turn of echoing
141 1393 1   BASS$ERT : NOVALUE, ! Return error message
142 1394 1   BASS$CB_PUSH : JSB CB PUSH NOVALUE, ! Load register CCB
143 1395 1   BASS$CB_POP : JSB CB POP NOVALUE, ! Done with register CCB
144 1396 1   BASS$STOP_IO : NOVALUE, ! Signal fatal I/O error
145 1397 1   BPAS$MESAG,        ! Do send/receive
146 1398 1   BPAS$ASSIGN,       ! Assign a device
147 1399 1   BPAS$DEASSIGN,     ! Deassign a device
148 1400 1   BPAS$DEASS ALL,    ! Deassign all devices
149 1401 1   BPAS$SET_DEF,      ! Set default
150 1402 1   BPAS$FSS,         ! File string scan
151 1403 1   BPAS$FREE BLOCK,   ! Free heap storage
152 1404 1   BPAS$SET_PRI;      ! Change priority
153 1405 1
154 1406 1 !+
155 1407 1 ! The following are the error codes used in this module.
156 1408 1 !-
157 1409 1
158 1410 1 EXTERNAL LITERAL
159 1411 1   BASS$K_ILLIO_CHA : UNSIGNED (8), ! Illegal I/O channel
160 1412 1   BASS$K_IO_CHANOT : UNSIGNED (8), ! I/O channel not open
161 1413 1   BASS$K_ILSYSUSA : UNSIGNED (8), ! Illegal SYSS() usage
162 1414 1   BASS$K_ILLBYTCOU : UNSIGNED (8), ! Illegal byte count for I/O
163 1415 1   BASS$K_NO ROOUSE : UNSIGNED (8), ! No room for user on device
164 1416 1   BASS$K_MISSPEFEA : UNSIGNED (8); ! Missing special feature
165 1417 1
```



```
167 1418 1 GLOBAL ROUTINE BAS$$SYS (
168 1419 1     RESULT_STR,
169 1420 1     CODE_STR
170 1421 1 ) : NOVALUE =
171 1422 1
172 1423 1 ++
173 1424 1 FUNCTIONAL DESCRIPTION:
174 1425 1
175 1426 1     Do a RSTS/E compatible SYS() function.
176 1427 1
177 1428 1 FORMAL PARAMETERS:
178 1429 1
179 1430 1     RESULT_STR.wz.dx     Result of the SYS() function
180 1431 1     CODE_STR.rz.dx       A string of bytes which specify what to do.
181 1432 1
182 1433 1 IMPLICIT INPUTS:
183 1434 1
184 1435 1     NONE
185 1436 1
186 1437 1 IMPLICIT OUTPUTS:
187 1438 1
188 1439 1     NONE
189 1440 1
190 1441 1 ROUTINE VALUE:
191 1442 1
192 1443 1     Depends on the function, see above. Where the result is not
193 1444 1     defined the null string is returned.
194 1445 1
195 1446 1 SIDE EFFECTS:
196 1447 1
197 1448 1     Signals if an error is encountered.
198 1449 1
199 1450 1 --
200 1451 1
201 1452 2 BEGIN
202 1453 2
203 1454 2 MAP
204 1455 2     CODE_STR : REF BLOCK [8, BYTE];
205 1456 2
206 1457 2 BIND
207 1458 2     FIRST_BYTE = CODE_STR [DSC$A_POINTER] : REF VECTOR [, BYTE],
208 1459 2     STR_LENGTH = .CODE_STR [DSC$W_LENGTH];
209 1460 2
210 1461 2 LOCAL
211 1462 2     RET_STRING : BLOCK [8, BYTE];
212 1463 2
213 1464 2     RET_STRING [DSC$W_LENGTH] = 0;
214 1465 2     RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
215 1466 2     RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
216 1467 2     RET_STRING [DSC$A_POINTER] = 0;
217 1468 2
218 1469 2 ++
219 1470 2 Dispatch on the first byte of the string.
220 1471 2
221 1472 2 IF (STR_LENGTH EQLU 0)
222 1473 2 THEN
223 1474 2     BAS$$STOP (BAS$K_ILLSYSUSA)
```

```
224 1475 2 ELSE
225 1476
226 1477 CASE .FIRST_BYTE [0] FROM 0 TO 12 OF
227 1478 SET
228 1479
229 1480 [0] : ! Cancel control 0
230 1481 BASSRCTRLD ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
231 1482
232 1483 [1] : ! Enter TAPE mode
233 1484 BASS$STOP (BASS$K_MISSPEFEA);
234 1485
235 1486 [2] : ! Enable echoing
236 1487 BAS$ECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
237 1488
238 1489 [3] : ! Disable echoing
239 1490 BAS$NOECHO ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
240 1491
241 1492 [4] : ! ODT submode
242 1493 BEGIN
243 1494 BASS$STOP (BASS$K_MISSPEFEA);
244 1495 BAS$ONECHR ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
245 1496 END;
246 1497
247 1498 [5] : ! Exit with no prompt
248 1499 $EXIT (CODE = SS$_NORMAL);
249 1500
250 1501 [6] : ! Call file processor
251 1502 BAS$$UUD (RET_STRING, .CODE_STR);
252 1503
253 1504 [7] : ! Get core common
254 1505 BEGIN
255 1506 LOCAL
256 1507 STATUS;
257 1508
258 1509 STATUS = LIB$GET_COMMON (RET_STRING);
259 1510
260 1511 IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
261 1512
262 1513 END;
263 1514
264 1515 [8] : ! Put core common
265 1516 BEGIN
266 1517 LOCAL
267 1518 STATUS,
268 1519 LEN,
269 1520 DESC : BLOCK [8, BYTE];
270 1521
271 1522 LEN = STR_LENGTH - 1;
272 1523
273 1524 IF (.LEN GTR 127) THEN LEN = 0;
274 1525
275 1526 DESC [DSC$_LENGTH] = .LEN;
276 1527 DESC [DSC$_DTYPE] = DSC$K_DTYPE_Z;
277 1528 DESC [DSC$_CLASS] = DSC$K_CLASS_S;
278 1529 DESC [DSC$_POINTER] = FIRST_BYTE [1];
279 1530
280 1531 3
```



```
281 1532
282 1533
283 1534
284 1535
285 1536
286 1537
287 1538
288 1539
289 1540
290 1541
291 1542
292 1543
293 1544
294 1545
295 1546
296 1547
297 1548
298 1549
299 1550
300 1551
301 1552
302 1553
303 1554
304 1555
305 1556
306 1557
```

```
STATUS = LIB$PUT_COMMON (DESC);
IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
END;
[9] :                               ! Exit and clear program
      $EXIT (CODE = SS$_NORMAL);
[10] :                               ! Special implementation
      BAS$$STOP (BAS$K_MISSPEFEA);
[11] :                               ! Cancel type ahead
      BAS$CANTYPAHEAD ((IF (STR_LENGTH GEQ 2) THEN .FIRST_BYTE [1] ELSE 0));
[12] :                               ! Info on last opened file
      BAS$$STOP (BAS$K_MISSPEFEA);
[OUTRANGE] :                         ! Not defined
      BAS$$STOP (BAS$K_ILLSYSUSA);
TES;
STR$COPY_DX (.RESULT_STR, RET_STRING);
STR$FREE_DX (RET_STRING);
RETURN;
END;                               ! of routine BAS$SYS
```

```
.TITLE BAS$SYS
.IDENT \1-014\
```

```
.EXTRN LIB$STOP, STR$COPY_DX
.EXTRN STR$COPY_R, STR$FREE1_DX
.EXTRN STR$CONCAT, LIB$GET_COMMON
.EXTRN LIB$PUT_COMMON, BAS$$STOP
.EXTRN BAS$RCTRL0, BAS$NOECHO
.EXTRN BAS$ONECHR, BAS$CANTYPAHEAD
.EXTRN BAS$CTRLC, BAS$ECHO
.EXTRN BAS$ERT, BAS$$CB_PUSH
.EXTRN BAS$$CB_POP, BAS$$STOP_IO
.EXTRN BPAS$MSG, BPAS$ASSIGN
.EXTRN BPAS$DEASSIGN, BPAS$DEASS_ALL
.EXTRN BPASSET_DEF, BPAS$FS
.EXTRN BPAS$FREE_BLOCK, BPASSET_PRI
.EXTRN BAS$K_ILC10_CHA
.EXTRN BAS$K_IO_CHA_NOT
.EXTRN BAS$K_ILC1SYSUSA
.EXTRN BAS$K_ILC1BYTCOU
.EXTRN BAS$K_NO_R0OUSE
.EXTRN BAS$K_MISSPEFEA
.EXTRN SYS$EXIT
```

```
.PSECT _BAS$CODE, NOWRT, SHR, PIC, 2
```

```
.ENTRY BAS$SYS, Save R2,R3,R4
MOVAB BAS$$STOP, R4
SUBL2 #16, SP
```

```
54 00000000G 001C 00000
5E          10 9E 00002
           10 C2 00009
```

```
: 1418
:
:
```

50	08	AC	04	C1	0000C	ADDL3	#4, CODE STR, R0	1458
		53	BC	3C	00011	MOVZWL	@CODE STR, R3	1459
	08	AE	8F	D0	00015	MOVL	#3355432, RET_STRING	1464
			AE	D4	0001D	CLRL	RET_STRING+4	1467
			53	D5	00020	TSTL	R3	1472
			21	13	00022	BEQL	2\$	
		52	60	D0	00024	MOVL	(R0), R2	1477
	OC	00	62	8F	00027	CASEB	(R2), #0, #12	
004D	0037	00EE	0021		0002B	.WORD	3\$-1\$,-	
008D	0080	00CD	0063		00033		27\$-1\$,-	
00D8	00EE	00CD	0099		0003B		6\$-1\$,-	
			00EE		00043		9\$-1\$,-	
							12\$-1\$,-	
							22\$-1\$,-	
							15\$-1\$,-	
							17\$-1\$,-	
							18\$-1\$,-	
							22\$-1\$,-	
							27\$-1\$,-	
							24\$-1\$,-	
							27\$-1\$	
	7E	00G	8F	9A	00045	MOVZBL	#BASSK_ILLSYSUSA, -(SP)	1551
			00D1	31	00049	BRW	28\$	
	02		53	B1	0004C	CMPW	R3, #2	1481
			06	1F	0004F	BLSSU	4\$	
	7E	01	A2	9A	00051	MOVZBL	1(R2), -(SP)	
			02	11	00055	BRB	5\$	
			7E	D4	00057	CLRL	-(SP)	
00000000G	00		01	FB	00059	CALLS	#1, BASSRCTRL0	
			54	11	00060	BRB	16\$	
	02		53	B1	00062	CMPW	R3, #2	1487
			06	1F	00065	BLSSU	7\$	
	7E	01	A2	9A	00067	MOVZBL	1(R2), -(SP)	
			02	11	0006B	BRB	8\$	
			7E	D4	0006D	CLRL	-(SP)	
00000000G	00		01	FB	0006F	CALLS	#1, BASSECHO	
			7E	11	00076	BRB	21\$	
	02		53	B1	00078	CMPW	R3, #2	1490
			06	1F	0007B	BLSSU	10\$	
	7E	01	A2	9A	0007D	MOVZBL	1(R2), -(SP)	
			02	11	00081	BRB	11\$	
			7E	D4	00083	CLRL	-(SP)	
00000000G	00		01	FB	00085	CALLS	#1, BASSNOECHO	
			73	11	0008C	BRB	23\$	
	7E	00G	8F	9A	0008E	MOVZBL	#BASSK_MISSPEFEA, -(SP)	1494
64			01	FB	00092	CALLS	#1, BASS\$STOP	
02			53	B1	00095	CMPW	R3, #2	1495
			06	1F	00098	BLSSU	13\$	
	7E	01	A2	9A	0009A	MOVZBL	1(R2), -(SP)	
			02	11	0009E	BRB	14\$	
			7E	D4	000A0	CLRL	-(SP)	
00000000G	00		01	FB	000A2	CALLS	#1, BASSONECHR	
			75	11	000A9	BRB	29\$	1477
		08	AC	DD	000AB	PUSHL	CODE STR	1502
		OC	AE	9F	000AE	PUSHAB	RET_STRING	
			02	FB	000B1	CALLS	#2, BASS\$UUO	
0000V	CF		68	11	000B6	BRB	29\$	

00000000G	00	08	AE	9F	000B8	17\$:	PUSHAB	RET_STRING	:	1510
			01	FB	000B8		CALLS	#1, LIB\$GET_COMMON	:	
			26	11	000C2		BRB	20\$	:	1512
	50	FF	A3	9E	000C4	18\$:	MOVAB	-1(R3), LEN	:	1524
0000007F	8F		50	D1	000C8		CMPL	LEN, #127	:	1526
			02	15	000CF		BLEQ	19\$	:	
			50	D4	000D1		CLRL	LEN	:	
	6E		50	B0	000D3	19\$:	MOVW	LEN, DESC	:	1528
02	AE	0100	8F	B0	000D6		MOVW	#256, DESC+2	:	1529
04	AE	01	A2	9E	000DC		MOVAB	1(R2), DESC+4	:	1531
			5E	DD	000E1		PUSHL	SP	:	1532
00000000G	00		01	FB	000E3		CALLS	#1, LIB\$PUT_COMMON	:	
	33		50	E8	000EA	20\$:	BLBS	STATUS, 29\$	:	1534
			50	DD	000ED		PUSHL	STATUS	:	
00000000G	00		01	FB	000EF		CALLS	#1, LIB\$STOP	:	
			28	11	000F6	21\$:	BRB	29\$	:	1477
			01	DD	000F8	22\$:	PUSHL	#1	:	1539
00000000G	00		01	FB	000FA		CALLS	#1, SYS\$EXIT	:	
			1D	11	00101	23\$:	BRB	29\$	:	
	02		53	B1	00103	24\$:	CMPL	R3, #2	:	1545
			06	1F	00106		BLSSU	25\$	:	
	7E	01	A2	9A	00108		MOVZBL	1(R2), -(SP)	:	
			02	11	0010C		BRB	26\$	:	
			7E	D4	0010E	25\$:	CLRL	-(SP)	:	
00000000G	00		01	FB	00110	26\$:	CALLS	#1, BAS\$CANTYPAHEAD	:	
			07	11	00117		BRB	29\$	:	
	7E	00G	8F	9A	00119	27\$:	MOVZBL	#BAS\$K_MISPEFEA, -(SP)	:	1548
	64		01	FB	0011D	28\$:	CALLS	#1, BAS\$\$STOP	:	
		08	AE	9F	00120	29\$:	PUSHAB	RET_STRING	:	1554
		04	AC	DD	00123		PUSHL	RESULT_STR	:	
00000000G	00		02	FB	00126		CALLS	#2, STR\$COPY_DX	:	
		08	AE	9F	0012D		PUSHAB	RET_STRING	:	1555
00000000G	00		01	FB	00130		CALLS	#1, STR\$FREE1_DX	:	
			04	00137		RET			:	1557

; Routine Size: 312 bytes, Routine Base: \_BAS\$CODE + 0000

; 307 1558 1



```
309 1559 1 ROUTINE BASS$UUD (
310 1560 1     RESULT_STR,
311 1561 1     CODE_STR
312 1562 1 ) : NOVALUE =
313 1563 1
314 1564 1
315 1565 1
316 1566 1
317 1567 1
318 1568 1
319 1569 1
320 1570 1
321 1571 1
322 1572 1
323 1573 1
324 1574 1
325 1575 1
326 1576 1
327 1577 1
328 1578 1
329 1579 1
330 1580 1
331 1581 1
332 1582 1
333 1583 1
334 1584 1
335 1585 1
336 1586 1
337 1587 1
338 1588 1
339 1589 1
340 1590 1
341 1591 1
342 1592 1
343 1593 1
344 1594 2
345 1595 2
346 1596 2
347 1597 2
348 1598 2
349 1599 2
350 1600 2
351 1601 2
352 1602 2
353 1603 2
354 1604 2
355 1605 2
356 1606 2
357 1607 2
358 1608 2
359 1609 2
360 1610 2
361 1611 2
362 1612 2
363 1613 2
364 1614 2
365 1615 2

ROUTINE BASS$UUD (
    RESULT_STR,
    CODE_STR
) : NOVALUE =

    Do a SYS(CHR$(6)) function
    Result of the SYS() call
    String that tells what to do

**
FUNCTIONAL DESCRIPTION:
    Do a RSTS/E compatible SYS() function, where the first byte of
    the string is a 6.

FORMAL PARAMETERS:
    RESULT_STR.wz.dx    The result of the SYS() function
    CODE_STR.rz.dx      A string of bytes which specify what to do.

IMPLICIT INPUTS:
    NONE

IMPLICIT OUTPUTS:
    NONE

ROUTINE VALUE:
    Depends on the function, see above. Where the result is not
    specified, the user's string is set to null.

SIDE EFFECTS:
    Signals if an error is encountered.

--
BEGIN
EXTERNAL ROUTINE
    RSOASC : NOVALUE;

MAP
    CODE_STR : REF BLOCK [8, BYTE];

LOCAL
    RET_STRING : BLOCK [8, BYTE],
    FIRQB : BLOCK [512 + 41 + 2, BYTE] FIELD (FQB$FIELDS),
    FIRQB_DESC : BLOCK [8, BYTE];

    FIRQB_DESC [DSC$W_LENGTH] = 512 + 41;
    FIRQB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
    FIRQB_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
    FIRQB_DESC [DSC$A_POINTER] = FIRQB [2, B_];
    CH$COPY (.CODE_STR [DSC$W_LENGTH], .CODE_STR [DSC$A_POINTER], 0, 512 + 41, FIRQB [2, B_]);
    RET_STRING [DSC$W_LENGTH] = 0;
    RET_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_Z;
    RET_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
    RET_STRING [DSC$A_POINTER] = 0;
```

```

1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672

```

```

+ Dispatch on the second byte of the string.
-
CASE .FIRQB [FQBSB_FUNCTION] FROM FUN$K_MINUUD TO FUN$K_MAXUUD OF
  SET
    [FUN$K_UUCT] : ! Control C trap enable
    BAS$CTRLC ();
    [FUN$K_UUERR] : ! Get error message
    BEGIN
      LOCAL
        ERR_STRING : BLOCK [8, BYTE],
        HEADER : BLOCK [8, BYTE];

        ERR_STRING [DSC$W_LENGTH] = 0;
        ERR_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
        ERR_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
        ERR_STRING [DSC$A_POINTER] = 0;
        BAS$ERT (ERR_STRING, .FIRQB [FQBSB_ERRNUM]);
        HEADER [DSC$W_LENGTH] = 2;
        HEADER [DSC$B_DTYPE] = DSC$K_DTYPE_BU;
        HEADER [DSC$B_CLASS] = DSC$K_CLASS_S;
        HEADER [DSC$A_POINTER] = UPLIT (BYTE (0, 0));
        STR$CONCAT (RET_STRING, HEADER, ERR_STRING);
        STR$FREE1_DX (ERR_STRING);
      END;

    [FUN$K_UUMES1] : ! Small message send/receive
    BEGIN
      LITERAL
        K_SML_SEND = -1,
        K_SML_REMREC = 0,
        K_SML_DCLREC_REC = 1,
        K_SML_REC = 2;

      LOCAL
        BYTXFR, ! Number of bytes actually transferred
        ASCII_LOGNAM : VECTOR [6, BYTE], ! A buffer to put the translated R50 name.
        RES_STRING : VECTOR [30, BYTE], ! Some place to put the result string temporarily.
        MESAG : VECTOR [20, BYTE]; ! A temp. place to put the message.

+ Translate the name passed to ASCII.
-
      R50ASC (%REF (6), FIRQB [FQBSB_RCVNAM], ASCII_LOGNAM);

      IF .ASCII_LOGNAM [0] EQL %C'?' THEN BAS$$$STOP_10 (BAS$K_ILLSYSUSA);

+ Do each function seperately
-
      CASE .FIRQB [FQBSB_SUBFUN] FROM K_SML_SEND TO K_SML_REC OF

```

```
SET
[K_SML_SEND] :
  BEGIN
    CHSMOVE (20, FIRQB [10, B ], MESAG);
    CHSMOVE (20, MESAG, FIRQB [12, B ]);      !The message
    CHSMOVE (6, ASCII_LOGNAM, FIRQB [FOBST_RCVNAM]); !Logical name
  END;

[K_SML_REMREC] :
  1;      ! Looks the same, so don't do anything.

[K_SML_DCLREC_REC] :
  BEGIN
    CHSMOVE (6, ASCII_LOGNAM, FIRQB [FOBST_RCVNAM]); !Logical name
    FIRQB [FOBSW_BMAX] = -1;      ! Use temporary mailboxes.
    FIRQB [FOBSB_ACCESS] = 1;    ! Local message.
    BPASMESAG (FIRQB, 0, 0, BYTXFR);
    CHSFILL (0, 32, FIRQB [2, B ]);
    FIRQB [FOBSB_SUBFUN] = K_SML_REC;
  END;

[K_SML_REC] :
  BEGIN
    CHSMOVE (6, ASCII_LOGNAM, FIRQB [FOBST_RCVNAM]); !Logical name
    FIRQB [FOBSB_SUBFUN] = K_SML_DCLREC_REC;
    FIRQB [FOBSW_BMAX] = -1;      ! Use temporary mailboxes.
    FIRQB [FOBSB_ACCESS] = 1;    ! Local message.
    BPASMESAG (FIRQB, 0, 0, BYTXFR);
    CHSFILL (0, 32, FIRQB [2, B ]);
    FIRQB [FOBSB_SUBFUN] = K_SML_REC;
    FIRQB [FOBSB_RMOD] = 1;      !Sleep indefinitely
  END;

TES;
```

```
Now call BPASMESAG, to do the work.
```

```
BPASMESAG (FIRQB, 0, 0, BYTXFR);
CHSFILL (0, 30, RES_STRING);
CHSMOVE (20, FIRQB [FOBST_PAR_STR], RES_STRING + 8);
STR$COPY_R (RES_STRING, XREF 730), RES_STRING);
END;
```

```
[FUN$K_UUMES2] :      ! Large message send/receive
  BEGIN
    GLOBAL REGISTER
      CCB = K_CCB_REG : REF BLOCK [, BYTE];

    LOCAL
      BUFLN,      ! Length of buffer
      BUFADR,     ! Address of user's buffer
      BYTXFR;     ! Number of bytes actually transferred
```

```
Set up buffer length, byte count and buffer address based on the SYS()
```



```
480 1730 3 1_string.
481 1731 3 1-
482 1732 3 1-
483 1733 4 IF (.FIRQB [12, B_] EQL 0)
484 1734 3 THEN
485 1735 4 BEGIN
486 1736 4 1+
487 1737 4 1- The buffer is in the string.
488 1738 4 1-
489 1739 4 CCB = 0;
490 1740 4
491 1741 5 IF (.CODE_STR [DSC$W_LENGTH] GTR 40)
492 1742 4 THEN
493 1743 5 BEGIN
494 1744 5 BUFADR = FIRQB [42, B_];
495 1745 5 BUFLN = .CODE_STR [DSC$W_LENGTH] - 40;
496 1746 5 END
497 1747 4 ELSE
498 1748 5 BEGIN
499 1749 5 1+
500 1750 5 1- There is no buffer.
501 1751 5 1-
502 1752 5 BUFADR = BUFLN = 0;
503 1753 5 END
504 1754 5
505 1755 4 END
506 1756 3 ELSE
507 1757 4 BEGIN
508 1758 4 1+
509 1759 4 1- The buffer is an I/O buffer. The low seven bits of byte 11 are the
510 1760 4 channel number.
511 1761 4 1-
512 1762 4
513 1763 4 LOCAL
514 1764 4 CHAN;
515 1765 4
516 1766 4 CHAN = (.FIRQB [12, B_] AND 127);
517 1767 4
518 1768 4 IF (.CHAN LEQ 0) THEN BASS$STOP_IO (BASS$ILLIO_CHA);
519 1769 4
520 1770 4 BASS$CB_PUSH (.CHAN, LUB$K_LUN_MIN);
521 1771 4
522 1772 4 IF ( NOT .CCB [LUB$V_OPENED]) THEN BASS$STOP_IO (BASS$IO_CHANOT);
523 1773 4
524 1774 5 IF ((.FIRQB [16, W_] + .FIRQB [14, W_]) GTRU .CCB [LUB$W_RBUF_SIZE])
525 1775 4 THEN
526 1776 4 BASS$STOP_IO (BASS$ILLBYTCOU);
527 1777 4
528 1778 4 IF (.FIRQB [14, W_] EQL 0) THEN BASS$STOP_IO (BASS$NO_ROOUSE);
529 1779 4
530 1780 4 BUFADR = .CCB [LUB$A_RBUF_ADR] + .FIRQB [16, W_];
531 1781 4 BUFLN = .FIRQB [14, W_];
532 1782 4 END;
533 1783 3
534 1784 3 1+
535 1785 3 1- Copy the user's parameter string to the FIRQB.
536 1786 3 1-
```

```
537 1787 CHSMOVE (20, FIRQB [22, B_], FIRQB [12, B_]);
538 1788
539 1789 + Now do the RSTS/E .MESAG function
540 1790
541 1791 BPASMESAG (FIRQB, .BUFLEN, .BUFADR, BYTXFR);
542 1792
543 1793 + Put things back.
544 1794
545 1795 CHSMOVE (20, FIRQB [12, B_], FIRQB [22, B_]);
546 1796 FIRQB [14, W_] = .BYTXFR;
547 1797
548 1798 IF (.CCB NEQA 0) THEN BASS$CB_POP ();
549 1799
550 1800 STRSCOPY_R (RET_STRING, XREF (40), .FIRQB_DESC [DSC$A_POINTER]);
551 1801 END;
552 1802
553 1803 [FUN$K_UUFSS1, FUN$K_UUFSS2] : ! File string scan
554 1804 BEGIN
555 1805
556 1806 LOCAL
557 1807 STATUS;
558 1808 FSB : $FSB_DEF;
559 1809
560 1810 STATUS = BPA$FSS (FIRQB, FSB, .CODE_STR [DSC$A_POINTER] + 2, .CODE_STR [DSC$W_LENGTH] - 2);
561 1811
562 1812 IF ( NOT .STATUS)
563 1813 THEN
564 1814 LIB$STOP (.STATUS)
565 1815 ELSE
566 1816 BEGIN
567 1817
568 1818 LOCAL
569 1819 STATUS;
570 1820
571 1821 STATUS = BPA$FREE_BLOCK (.FSB [FSB$A_FSA], NAM$C_MAXRSS);
572 1822
573 1823 IF ( NOT .STATUS) THEN LIB$STOP (.STATUS);
574 1824
575 1825 + Return information to the user from the FIRQB and FSB.
576 1826
577 1827
578 1828 FIRQB [3, B_] = .FIRQB [5, B_];
579 1829 FIRQB [16, W_] = .FIRQB [28, W_];
580 1830 FIRQB [20, W_] = .FIRQB [30, W_];
581 1831 FIRQB [28, W_] = .FSB [FSB$W_FLAG_1];
582 1832 FIRQB [30, W_] = .FSB [FSB$W_FLAG_2];
583 1833 STRSCOPY_R (RET_STRING, XREF (30), .FIRQB_DESC [DSC$A_POINTER]);
584 1834 END;
585 1835
586 1836 END;
587 1837
588 1838 [FUN$K_UUPRI] : ! Set priority, etc.
589 1839
590 1840 + Only priority setting is implemented; all else is ignored.
591 1841
592 1842 BEGIN
593 1843 BPA$SET_PRI (FIRQB);
```

```
594      1844 2
595      1845
596      1846
597      1847
598      1848
599      1849
600      1850
601      1851
602      1852
603      1853
604      1854
605      1855
606      1856
607      1857
608      1858
609      1859
610      1860
611      1861
612      1862
613      1863
614      1864
615      1865
616      1866
617      1867
618      1868
619      1869
620      1870
621      1871
622      1872
623      1873
624      1874
625      1875
626      1876
627      1877
628      1878 1

      END;

[FUN$K UUATR] :      ! Read/write file
      BAS$$STOP (BAS$K_MISSPEFEA);

[FUN$K UUASS] :      ! Assign
      BEGIN
      BPAS$ASSIGN (FIRQB);
      STR$COPY_DX (RET_STRING, FIRQB_DESC);
      END;

[FUN$K UUDEA] :      ! Deassign
      BEGIN
      BPAS$DEASSIGN (FIRQB);
      END;

[FUN$K UUDAL] :      ! Deassign all
      BPAS$DEASS_ALL ();

[FUN$K UUSDEF] :      ! Set default
      BEGIN
      BPAS$SET_DEF (FIRQB);
      END;

[INRANGE] :      ! Unimplemented
      BAS$$STOP (BAS$K_MISSPEFEA);

[OUTRANGE] :      ! Not defined
      BAS$$STOP (BAS$K_ILLSYSUSA);
      TES;

STR$COPY_DX (.RESULT_STR, RET_STRING);
STR$FREE_T_DX (RET_STRING);
RETURN;
      END;      ! of routine BAS$$UUO
```

00 00 00138 P.AAA: .BYTE 0, 0

.EXTRN R50ASC

OFFC 00000 BAS\$\$UUO:

		5A	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1559	
		59	00000000G	00	9E	00009	MOVAB	BPAS\$MESAG, R10		
		5E	FD7C	CE	9E	00010	MOVAB	BAS\$\$STOP_10, R9		
48		AE	01020229	8F	D0	00015	MOVAB	-644(SP), SP	1607	
4C		AE	52	AE	9E	0001D	MOVL	#16908841, FIRQB_DESC	1610	
		56	08	AC	D0	00022	MOVAB	FIRQB+2, FIRQB_DESC+4	1611	
0229	8F	04	B6	66	2C	00026	MOVL	CODE_STR, R6		
			52	AE		0002E	MOVCS	(R6), @4(R6), #0, #553, FIRQB+2		
		F8	AD	02000000	8F	D0	00030	MOVL	#33554432, RET_STRING	1612
			FC	AD	D4	00038	CLRL	RET_STRING+4	1615	
		E6	8F	53	AE	8F	0003B	CASEB	FIRQB+3, #-26, #54	1620
0214			02C9	02C9		00041	1\$:	.WORD	32\$-1\$,-	
02C9			02C9	02C9		00049			32\$-1\$,-	



Page 16  
(4)[illegible]

MOVZBL 315-15  
BRW #BAS\$K\_ILL\$SYSUSA, -(SP)  
CALLS 33\$  
BRB #0, BAS\$CTRLC  
4\$

1872  
1624

40	AE	020E0700	8F	D0	000BF	3\$:	MOVL	#34471936, ERR_STRING	1633
		44	AE	D4	000C7		CLRL	ERR_STRING+4	1636
	7E	54	AE	9A	000CA		MOVZBL	FIRQB+4, -(SP)	1637
		44	AE	9F	000CE		PUSHAB	ERR_STRING	
00000000G	00		02	FB	000D1		CALLS	#2, BASSERT	
38	AE	01020002	8F	D0	000D8		MOVL	#16908290, HEADER	1638
3C	AE	FF1A	CF	9E	000E0		MOVAB	P.AAA, HEADER+4	1641
		40	AE	9F	000E6		PUSHAB	ERR_STRING	1642
		3C	AE	9F	000E9		PUSHAB	HEADER	
		F8	AD	9F	000EC		PUSHAB	RET_STRING	
00000000G	00		03	FB	000EF		CALLS	#3, STR\$CONCAT	
		40	AE	9F	000F6		PUSHAB	ERR_STRING	1643
00000000G	00		01	FB	000F9		CALLS	#1, STR\$FREE1_DX	
		0212	31	00100	4\$:	BRW	34\$		1620
		40	AE	9F	00103	5\$:	PUSHAB	ASCII_LOGNAM	1664
		5A	AE	9F	00106		PUSHAB	FIRQB+6	
08	AE		06	D0	00109		MOVL	#6, 8(SP)	
		08	AE	9F	0010D		PUSHAB	8(SP)	
00000000G	00		03	FB	00110		CALLS	#3, R50ASC	
		3F	40	AE	91	00117	CMPB	ASCII_LOGNAM, #63	1666
		7E	07	12	0011B		BNEQ	6\$	
		69	00G	8F	9A	0011D	MOVZBL	#BASSK_ILLSYSUSA, -(SP)	
		8F	01	FB	00121		CALLS	#1, BASS\$STOP_10	
0042	03	FF	54	AE	8F	00124	CASEB	FIRQB+4, #-1, #3	1672
	001C	006C	0008		0012A	7\$:	.WORD	8\$-7\$, -	
								11\$-7\$, -	
								9\$-7\$, -	
								10\$-7\$, -	
OC	AE	5A	AE	14	28	00132	MOV C3	#20, FIRQB+10, MESAG	1677
5C	AE	OC	AE	14	28	00138	MOV C3	#20, MESAG, FIRQB+12	1678
56	AE	40	AE	06	28	0013E	MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1679
				50	11	00144	BRB	11\$	1672
56	AE	40	AE	06	28	00146	MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1687
		5E	AE	01	AE	0014C	MNEG W	#1, FIRQB+14	1688
		5D	AE	01	90	00150	MOV B	#1, FIRQB+13	1689
			04	AE	9F	00154	PUSHAB	BYTXFR	1690
				7E	7C	00157	CLRQ	-(SP)	
			5C	AE	9F	00159	PUSHAB	FIRQB	
		6A		04	FB	0015C	CALLS	#4, BPASMESAG	
20	00	6E		00	2C	0015F	MOV C5	#0, (SP), #0, #32, FIRQB+2	1691
			52	AE		00164			
		54	AE	02	90	00166	MOV B	#2, FIRQB+4	1692
			2A	11	0016A		BRB	11\$	1672
56	AE	40	AE	06	28	0016C	MOV C3	#6, ASCII_LOGNAM, FIRQB+6	1697
		54	AE	01	90	00172	MOV B	#1, FIRQB+4	1698
		5E	AE	01	AE	00176	MNEG W	#1, FIRQB+14	1699
		5D	AE	01	90	0017A	MOV B	#1, FIRQB+13	1700
			04	AE	9F	0017E	PUSHAB	BYTXFR	1701
				7E	7C	00181	CLRQ	-(SP)	
			5C	AE	9F	00183	PUSHAB	FIRQB	
		6A		04	FB	00186	CALLS	#4, BPASMESAG	
20	00	6E		00	2C	00189	MOV C5	#0, (SP), #0, #32, FIRQB+2	1702
			52	AE		0018E			
		54	AE	0102	8F	B0	MOV W	#258, FIRQB+4	1703
			04	AE	9F	00196	PUSHAB	BYTXFR	1711
				7E	7C	00199	CLRQ	-(SP)	
			5C	AE	9F	0019B	PUSHAB	FIRQB	

1E	00	6A	04	FB	0019E	CALLS	#4, BP\$MESAG	1712		
		6E	00	2C	001A1	MOV C5	#0, (SP), #0, #30, RES_STRING			
	28	AE	14	28	001A6	MOV C3	#20, FIRQB+12, RES_STRING+8	1713		
		5C	AE	9F	001AE	PUSHAB	RES_STRING	1714		
			00FD	31	001B1	BRW	24\$			
			5C	AE	95	001B4	12\$: TSTB	FIRQB+12	1733	
				17	12	001B7	BNEQ	14\$		
				5B	D4	001B9	CLRL	CCB	1739	
		28		66	B1	001BB	CMPW	(R6), #40	1741	
				0C	1B	001BE	BLEQU	13\$		
		57	7A	AE	9E	001C0	MOVAB	FIRQB+42, BUFADR	1744	
		58		66	3C	001C4	MOVZWL	(R6), BUFLN	1745	
		58		28	C2	001C7	SUBL2	#40, BUFLN		
				58	11	001CA	BRB	19\$	1741	
				57	7C	001CC	13\$: CLRQ	BUFADR	1752	
				54	11	001CE	BRB	19\$	1741	
52	5C	AE	07	00	EF	001D0	14\$: EXTZV	#0, #7, FIRQB+12, CHAN	1766	
			7E	07	14	001D6	BGTR	15\$	1768	
			69	00G	8F	9A	001D8	MOVZBL	#BAS\$K_ILLIO_CHA, -(SP)	
				01	FB	001DC	CALLS	#1, BAS\$\$STOP_IO		
				50	D4	001DF	15\$: CLRL	R0	1770	
				00000000G	00	16	001E1	JSB	BAS\$\$CB_PUSH	
			07	FC	AB	E8	001E7	BLBS	-4(CCB), 16\$	1772
			7E	00G	8F	9A	001EB	MOVZBL	#BAS\$K_IO_CHANOT, -(SP)	
			69	01	FB	001EF	CALLS	#1, BAS\$\$STOP_IO		
			50	60	AE	3C	001F2	16\$: MOVZWL	FIRQB+16, R0	1774
			51	5E	AE	3C	001F6	MOVZWL	FIRQB+14, R1	
50	D2	AB	50		51	C0	001FA	ADDL2	R1, R0	
			10		00	ED	001FD	CMPZV	#0, #16, -46(CCB), R0	
				07	1E	00203	BGEQU	17\$		
			7E	00G	8F	9A	00205	MOVZBL	#BAS\$K_ILLBYTCOU, -(SP)	1776
			69	01	FB	00209	CALLS	#1, BAS\$\$STOP_IO		
				5E	AE	B5	0020C	17\$: TSTW	FIRQB+14	1778
				07	12	0020F	BNEQ	18\$		
			7E	00G	8F	9A	00211	MOVZBL	#BAS\$K_NO_R0OUSE, -(SP)	
			69	01	FB	00215	CALLS	#1, BAS\$\$STOP_IO		
			57	60	AE	3C	00218	18\$: MOVZWL	FIRQB+16, BUFADR	1780
			57	EC	AB	C0	0021C	ADDL2	-20(CCB), BUFADR	
			58	5E	AE	3C	00220	MOVZWL	FIRQB+14, BUFLN	1781
	5C	AE	66	AE	14	28	00224	19\$: MOV C3	#20, FIRQB+22, FIRQB+12	1787
				08	AE	9F	0022A	PUSHAB	BYTXFR	1791
				57	DD	0022D	PUSHL	BUFADR		
				58	DD	0022F	PUSHL	BUFLN		
			5C	AE	9F	00231	PUSHAB	FIRQB		
				04	FB	00234	CALLS	#4, BP\$MESAG		
	66	AE	5C	AE	14	28	00237	MOV C3	#20, FIRQB+12, FIRQB+22	1795
			5E	AE	AE	B0	0023D	MOVW	BYTXFR, FIRQB+14	1796
					5B	D5	00242	TSTL	CCB	1798
					06	13	00244	BEQL	20\$	
				00000000G	00	16	00246	JSB	BAS\$\$CB_POP	
				4C	AE	DD	0024C	20\$: PUSHL	FIRQB_DESC+4	1800
			04	AE	28	D0	0024F	MOVL	#40, 4(SP)	
					60	11	00253	BRB	25\$	
			7E		66	3C	00255	21\$: MOVZWL	(R6), -(SP)	1810
			6E		02	C2	00258	SUBL2	#2, (SP)	
	7E	04	A6		02	C1	0025B	ADDL3	#2, 4(R6), -(SP)	



		2C	AE	9F	00260	PUSHAB	FSB		
		5C	AE	9F	00263	PUSHAB	FIRQB		
00000000G	00		04	FB	00266	CALLS	#4, BPAS\$FSS		
	0B		50	E8	0026D	BLBS	STATUS, 22\$		1812
00000000G	00		50	DD	00270	PUSHL	STATUS		1814
			01	FB	00272	CALLS	#1, LIB\$STOP		
	7E		78	11	00279	BRB	29\$		
		FF	8F	9A	0027B	MOVZBL	#255, -(SP)		1821
		2C	AE	DD	0027F	PUSHL	FSB+4		
00000000G	00		02	FB	00282	CALLS	#2, BPAS\$FREE_BLOCK		
	09		50	E8	00289	BLBS	STATUS, 23\$		1823
			50	DD	0028C	PUSHL	STATUS		
00000000G	00		01	FB	0028E	CALLS	#1, LIB\$STOP		
	53	55	AE	90	00295	MOVB	FIRQB+5, FIRQB+3		1828
	60	6C	AE	B0	0029A	MOVW	FIRQB+28, FIRQB+16		1829
	64	6E	AE	B0	0029F	MOVW	FIRQB+30, FIRQB+20		1830
	6C	46	AE	B0	002A4	MOVW	FSB+34, FIRQB+28		1831
	6E	44	AE	B0	002A9	MOVW	FSB+32, FIRQB+30		1832
		4C	AE	DD	002AE	PUSHL	FIRQB_DESC+4		1833
	04		1E	DD	002B1	MOVL	#30, 4(SP)		
		04	AE	9F	002B5	PUSHAB	4(SP)		
		F8	AD	9F	002B8	PUSHAB	RET_STRING		
00000000G	00		03	FB	002BB	CALLS	#3, STR\$COPY_R		
			51	11	002C2	BRB	34\$		1620
		50	AE	9F	002C4	PUSHAB	FIRQB		1843
00000000G	00		01	FB	002C7	CALLS	#1, BPAS\$SET_PRI		
			45	11	002CE	BRB	34\$		1620
		50	AE	9F	002D0	PUSHAB	FIRQB		1851
00000000G	00		01	FB	002D3	CALLS	#1, BPAS\$ASSIGN		
		48	AE	9F	002DA	PUSHAB	FIRQB_DESC		1852
		F8	AD	9F	002DD	PUSHAB	RET_STRING		
00000000G	00		02	FB	002E0	CALLS	#2, STR\$COPY_DX		
			2C	11	002E7	BRB	34\$		1620
		50	AE	9F	002E9	PUSHAB	FIRQB		1857
00000000G	00		01	FB	002EC	CALLS	#1, BPAS\$DEASSIGN		
			20	11	002F3	BRB	34\$		1620
00000000G	00		00	FB	002F5	CALLS	#0, BPAS\$DEASS_ALL		1861
			17	11	002FC	BRB	34\$		
		50	AE	9F	002FE	PUSHAB	FIRQB		1865
00000000G	00		01	FB	00301	CALLS	#1, BPAS\$SET_DEF		
			0B	11	00308	BRB	34\$		1620
	7E	00G	8F	9A	0030A	MOVZBL	#BAS\$K_MISSPEFEA, -(SP)		1869
00000000G	00		01	FB	0030E	CALLS	#1, BAS\$\$\$STOP		
		F8	AD	9F	00315	PUSHAB	RET_STRING		1875
		04	AC	DD	00318	PUSHL	RESULT_STR		
00000000G	00		02	FB	0031B	CALLS	#2, STR\$COPY_DX		
		F8	AD	9F	00322	PUSHAB	RET_STRING		1876
00000000G	00		01	FB	00325	CALLS	#1, STR\$FREE1_DX		
			04	0032C	RET				1878

; Routine Size: 813 bytes, Routine Base: \_BAS\$CODE + 013A

: 629 1879 1 END  
: 630 1880 1  
: 631 1881 0 ELUDOM

! of module BAS\$SYS

PSECT SUMMARY

Name	Bytes	Attributes
_BASSCODE	1127	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	13	0	581	00:01.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:BASSYS/OBJ=OBJ\$:BASSYS MSRC\$:BASSYS/UPDATE=(ENH\$:BASSYS)

: Size: 1125 code + 2 data bytes  
: Run Time: 00:25.7  
: Elapsed Time: 00:58.6  
: Lines/CPU Min: 4388  
: Lexemes/CPU-Min: 31558  
: Memory Used: 288 pages  
: Compilation Complete



0032 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

